

Le serveur Apache

Abdelali SAIDI

abdelali.saidi@gmail.com

Plan

- 1 Présentation
- 2 Le protocole HTTP
- 3 Configuration du serveur
- 4 Configuration avancée

Plan

- 1 Présentation
- 2 Le protocole HTTP
- 3 Configuration du serveur
- 4 Configuration avancée

Présentation

Le Web

- Le Web est la manière la plus simple d'utiliser Internet
- Une URL (Uniform Resource Locator) permet au navigateur d'indiquer l'emplacement d'un document voulu
- Le HTML (HyperText Markup Language) est un langage de présentation de documents.
- Le HTTP (Hypertext Transfer Protocol) est le protocole qui transfère les documents sur le Web.

Présentation

Les URLs

La syntaxe `<protocole>://<adresse_du_serveur>[:port]/<chemin_du_doc>`.

Exemples :

- `http://httpd.apache.org`
- `http://httpd.apache.org/docs/2.2/howto/htaccess.html`
- `http://httpd.apache.org:80/docs/2.2/howto/htaccess.html`
- `https://httpd.apache.org/docs`
- `ftp://httpd.apache.org/docs`

Clients Web

- curl, wget, lynx, Firefox ...

Plan

- 1 Présentation
- 2 **Le protocole HTTP**
- 3 Configuration du serveur
- 4 Configuration avancée

Le protocole HTTP

Présentation

- Le protocole HTTP se base sur TCP
- De sa version 1.0, chaque requête correspondait à une transaction TCP complète
- Ce n'est plus le cas avec la version 1.1 (Une transaction TCP pour plusieurs requêtes)
- Il supporte l'authentification
- Les principales requêtes:
 - GET : demande le transfert d'un document
 - HEAD : demande des informations sur une ressource (si la page demandé a été modifiée ou pas)
 - POST : renvoie au serveur les informations remplies dans un formulaire

Le protocole HTTP

Les codes de retours

Les principaux codes de retours:

- 200 : OK
- 301 : Le document a été déplacé définitivement
- 302 : Le document a été déplacé temporairement
- 303 : Il faut s'adresser à un autre serveur
- 304 : If-Modified-Since indiqué avec la requête GET
- 400 : La syntaxe de la requête est mauvaise
- 401 : La requête requiert une autorisation

Le protocole HTTP

Les codes de retours

Les principaux codes de retours:

- 403 : La ressource demandée est interdite
- 404 : La ressource demandée n'existe pas
- 407 : Authentification proxy requise
- 408 : Temps épuisé
- 500 : Erreur interne
- 501 : Requête non supportée
- 503 : Service non présent
- 504 : Le serveur est occupé

Le protocole HTTP

Les ports d'écoute

- 80/tcp : HTTP
- 443/tcp : HTTPS

Les cookies

Une cookie est une donnée stockée sur le navigateur envoyée par le serveur Web. Lorsque le client envoie une requête vers le serveur, il envoie également ses cookies

Plan

- 1 Présentation
- 2 Le protocole HTTP
- 3 Configuration du serveur**
- 4 Configuration avancée

Configuration du serveur

Le fichier httpd.conf

Apache utilise un seul fichier de configuration : `/etc/apache/httpd.conf`
(`/etc/apache2/apache2.conf`)

Ce fichier se compose de deux grandes parties :

- L'environnement du serveur : contient toutes les instructions pour la modification des paramètres du serveur (nbr max des clients à traiter simultanément, le répertoire racine du serveur, working mode, adresse et port d'écoute ...)
- La configuration du serveur : contient les ressources qui seront envoyé vers les clients

Configuration du serveur

Les instructions de l'environnement du serveur

- ServerRoot : Répertoire racine du serveur
- PidFile : Contient le PID de apache
- User : L'utilisateur utilisé par apache lorsqu'il a besoin des privilège root
- Group : Le groupe de User

Configuration du serveur

Gestion des connexions

- Timeout (en seconde)
- KeepAlive (on, off) : Accepter ou non plus qu'une requête dans une connexion
- MaxKeepAliveRequests (O pour l'illimité) : Nombre de requêtes maximal d'une connexion
- KeepAliveTimeout : La durée d'attente d'une requête suivante dans une même connexion
- MaxClients : Le nombre maximal des clients
- Listen : L'adresse et le numéro de port d'écoute
 - Si l'adresse n'est pas spécifié, le serveur fera l'écoute sur toutes ses interfaces
 - Exemples
Listen 80
Listen 192.168.1.1:8080

Configuration du serveur

Information personnel

Ces informations sont envoyées avec toutes les réponses

- ServerAdmin: Contient l'adresse de l'administrateur du serveur
- ServerName : Contient le nom du serveur

Configuration du serveur

Location des documents

Quand apache reçoit une requête, il a besoin de la localisation par default des documents

- DocumentRoot: spécifie le répertoire racine sur lequel apache fera la recherche de la ressource demandée
- DirectoryIndex : spécifie les fichiers à envoyer au cas où apache ne trouve pas la ressource demandée

Configuration du serveur

Les permissions sur les ressources

Le modèle de sécurité par default du serveur apache est définit dans les balises Directory. La syntaxe:

```
<Directory /chemin/du/répertoire>  
[ Instructions ]  
</Directory>
```

Recommandations

- Les répertoires héritent les propriétés d'accès de leur répertoire parent
- Commencer par les restrictions les plus strictes sur le répertoire racine et définir par la suite les permissions des sous-répertoires

Configuration du serveur

Les permissions sur les ressources

Les instructions :

- Options : Définit les fonctionnalités disponibles pour un répertoire particulier
- Order : détermine l'ordre dans lequel les directives Allow et Deny seront évaluées
- Allow/Deny from argument
- AllowOverride : définit comment sont gérés les fichiers .htaccess de ce répertoire
- Require : précise qu'il faut un compte dans le fichier de mots de passe pour accéder au répertoire

Configuration du serveur

Les permissions sur les ressources

La directive options suit la syntaxe Options [+/-]argument [+/-]argument ...
Suivant, une liste des options possibles:

- ExecCGI : L'exécution de scripts CGI est permise
- FollowSymLinks : Autorise à suivre les liens symboliques.
- Indexes : Permet de lister le contenu du répertoire si index.html est absent
- MultiViews : Autorise les vue multiples suivant un contexte
- All : Toutes les options sauf MultiViews
- None : Aucune option

Configuration du serveur

Les permissions sur les ressources

Exemple:

```
<Directory /usr/www>
```

```
Options Indexes -ExecCGI
```

```
Order allow,deny
```

```
Allow from 172.16 172.16.0.0/255.255.0.0 172.16.0.0/16
```

```
Deny from All
```

```
</Directory>
```

Configuration du serveur

Pages des erreurs

En cas d'erreur ou de problème, grâce à la directive *ErrorDocument* `<error number> <document>`, apache peut faire une des quatre actions suivantes:

- default : Afficher le message d'erreur par default
- Un message entre guillemets : un message personnalisé
- Un chemin vers un fichier local : Rediriger vers un document local
- Une URL : Rediriger vers une URL externe

Plan

- 1 Présentation
- 2 Le protocole HTTP
- 3 Configuration du serveur
- 4 Configuration avancée**

Configuration avancée

Authentification, autorisations et access d'accès

Pour demander aux clients de s'identifier avant de se connecter aux ressources, on peut utiliser les directives suivantes selon notre besoin:

- **AuthType** : spécifie le mode d'authentification (basic ou digest)
- **AuthName** : une chaîne de caractère qui sera affichée lors de l'authentification
- **AuthUserFile** : indique l'emplacement du fichier qui contient les mots de passe
- **AuthGroupFile** : indique le groupe auquel les clients doivent appartenir
- **Require** : spécifie les utilisateurs et les groupes autorisés:
 - `Require group groupe1 groupe2 ...`
 - `Require user user1 user2 ...`
 - `Require valid-user` (les comptes qui se trouvent sur `/etc/passwd`)

Configuration avancée

Le protocole SSL

Le SSL (Secure Socket Layer) permet de chiffrer la communication entre le serveur Apache et ses clients. Les directives qui permettent son activation :

- `SSLEngine` (on ou off) : spécifie si le protocole doit être utilisé ou pas
- `SSLCertificateFile` : indique l'emplacement du certificat numérique du serveur
- `SSLCertificateKeyFile` : indique l'emplacement de la clé privée associée au certificat
- `SSLRequireSSL` : permet de refuser la connexion si la communication n'est pas chiffrée

Configuration avancée

Les virtualhosts

Les virtualhosts permettent d'avoir plusieurs sites web sur une même machine. Pour cela, apache peut assurer selon l'une des deux techniques suivantes :

- selon l'adresse IP sollicité
- selon le nom de domaine sollicité

Configuration avancée

Les virtualhosts

Par adresse IP

On doit s'assurer que le serveur fait l'écoute sur toutes les adresses IP qu'on va utiliser. La syntaxe est la suivante:

```
<VirtualHost [ IP ou FQDN ]>  
[ paramètres ]  
</VirtualHost>
```

Les paramètres

ServerAdmin : L'adresse email de l'administrateur du serveur

DocumentRoot : L'emplacement du répertoire racine pour le site

ServerName : Nom du serveur

ErrorLog : Le fichier de journalisation du serveur

TransferLog : Toutes les requêtes vers ce site seront enregistrées dans ce fichier

Configuration avancée

Les virtualhosts

Par nom

Cette fois, la machine qui héberge les sites Web n'a qu'une seule adresse IP.
Exemple :

```
NameVirtualHost *:80
<VirtualHost *:80>
ServerName www.labo-linux.org
ServerAlias labo-linux.org *.labo-linux.org
DocumentRoot /var/www/labo-linux/
</VirtualHost>
```

Les paramètres

NameVirtualHost : spécifie l'adresse IP sur laquelle apache doit servir ses clients (la même que celle qui se trouve sur VirtualHosts)

ServerAlias : alias du nom du serveur